

Streamlit Essentials

From basics to advanced data app development

Surabhi Pandey



www.bpbonline.com

First Edition 2025

Copyright © BPB Publications, India

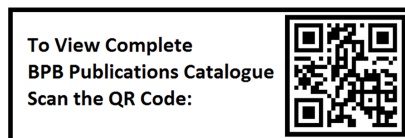
ISBN: 978-93-65890-822

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.



Dedicated to

My partner, the fundamental constant in my equation

About the Author

Surabhi is a data enthusiast with over a decade of experience across various data-focused roles. She holds a Bachelor's Degree in Computer Science from Amrita Vishwavidyapeetham. She enjoys the challenges and opportunities that come with turning raw data into meaningful insights and creating data-driven solutions that empower business decision-making.

Her background includes a mix of programming languages and tools; her current favorite toolkit includes DBT, Airflow, and Great Expectations, and her go-to language is Python. She has a keen interest in building frameworks to improve and streamline data quality and governance. Currently based in Kuala Lumpur, Surabhi enjoys a good cup of coffee and is an avid reader of epic fantasy novels.

About the Reviewers

- ❖ **Afaque** is an enthusiastic machine learning engineer who has gained experience in data science, AI, and the advancing fields of MLOps and GenAI. He approaches his work with curiosity and a willingness to learn, utilizing various tools and technologies. His current focus is on GenAI projects, emphasizing deployment, scaling, latency optimization, and monitoring. Outside of his professional role, Afaque shares insights and experiences through his Substack and Medium platforms, exploring various aspects of the data science and AI realm.
- ❖ **Vishu Aasliya** is an emerging tech enthusiast with a strong passion for machine learning and artificial intelligence. She has extensive experience in Docker and DevOps, focusing on developing and automating solutions that enhance efficiency and reduce manual intervention. Alongside her academic pursuits, Vishu is also a freelance developer, where she customizes web platforms and implements solutions to optimize user experience and drive engagement. She has worked on various projects, gaining hands-on experience in programming, containerization technologies, and process automation. Currently pursuing a degree in computer science with a specialization in artificial intelligence, Vishu is always eager to explore new technologies and broaden her skill set. In her free time, she enjoys exploring new tools, contributing to open-source projects, and staying updated with the latest advancements in tech.
- ❖ **Smaranjit Ghose** is a freelance machine learning engineer and lifelong learner specializing in data and algorithms to build fantastic tools and scalable applications. He is currently focused on developing distributed systems for applications serving millions of users, with particular emphasis on computer vision and **natural language processing (NLP)** implementations. Smaranjit is passionate about creating efficient, automated solutions and avoiding manual work wherever possible.

As a technical reviewer, he has contributed to various IT books covering machine learning, React, TensorFlow, **large language models (LLMs)**, and Python and JavaScript ecosystems. Beyond his professional endeavors, Smaranjit is an avid history enthusiast and UFC fan.

Smaranjit is also dedicated to mentoring, having guided over 10,000 students across India, the US, UAE, Ireland, and Mexico in building and contributing to open-source organizations and research labs. His commitment to education and open-source development reflects his belief in collaborative innovation and knowledge sharing in the tech community.

Acknowledgement

I would like to express my sincere gratitude to everyone who contributed to completing this book.

First and foremost, I extend my heartfelt appreciation to my family for their unwavering support and encouragement; their love and encouragement have been a constant source of motivation.

I would also like to thank the BPB Publications team for their excellent work. I would also like to acknowledge the reviewers and technical experts who provided valuable feedback and contributed to the refinement of this manuscript. Their insights and suggestions have significantly enhanced the quality of the book.

Some people leave a lasting impression when they cross your path. I had the pleasure of working with Dennis Lau and Raymond Au Yong a few years back. Dennis introduced me to Streamlit (and many other tools), reigniting my spark for experimentation. Raymond was always encouraging and supportive of the ideas I brought to the table, offering valuable feedback that refined my ideas and improved my design thinking. Their belief and their expertise helped me build confidence and trust in my abilities. Their influence still resonates in my tech journey today, and I am grateful to have crossed paths with them.

Last but not least, we want to express our gratitude to the readers who have shown interest in our book. Your support and encouragement have been deeply appreciated.

Thank you to everyone who has played a part in making this book a reality.

Preface

“**Streamlit Essentials: From basics to advanced data app development**” provides a comprehensive introduction to Streamlit, a powerful framework that simplifies the process of building and deploying data apps.

The chapters in this book take readers from the fundamentals of Streamlit to advanced techniques. It begins with the basics, covering **Introduction to Streamlit and Getting Started with Streamlit**, ensuring that the readers understand the platform’s needs and core concepts.

As you progress, you will explore the intricacies of **Layouts and Widgets** and how to integrate data to create visually appealing **Data Visualizations**. With a focus on real-world applications, you will also learn about **Machine Learning Model Deployments**.

The later chapters focus on more advanced topics, such as **Advanced Streamlit Concepts** and **Deploying Streamlit Apps**, helping you leverage Streamlit’s features like secret management, callbacks, Database connectivity, App testing and streamlining your workflow to make your application production-ready. In addition, hands-on projects in the book will enable you to showcase your skills by building a professional portfolio.

Building and Enhancing your Portfolio will help to use the power of networking and community building to get the most out of Streamlit and create a portfolio that resonates with potential employers and collaborators. **Enhancing Streamlit development with AI Tools** introduces AI-assisted development tools that can help speed up the process of application building from ideation to deployment.

This book is designed for data professionals, from beginners looking to explore Streamlit for the first time to advanced users seeking to optimize their development process. With practical examples and hands-on projects, “**Streamlit Mastery**” is a valuable resource for anyone aiming to include Streamlit in their data tooling arsenal and build interactive, data-driven applications.

Chapter 1: Introduction to Streamlit: This chapter discusses the challenges of data presentation and how this gap can be bridged. The most commonly used tools and also the challenges faced by the data developers while communicating and understanding the non-technical stakeholders. We discuss various challenges and explore why Streamlit fits in as an ideal solution for faster prototyping and custom data application development.

Chapter 2: Getting Started with Streamlit: This chapter covers the installation, basic setup, and core concepts that make Streamlit so accessible. Here, readers will gain strong foundation of Streamlit's working and understand how to get Streamlit up and running quickly. We will also dive into building the first interactive Data application, which we will enhance throughout future chapters.

Chapter 3: Exploring Streamlit Widgets: This chapter covers Streamlit-provided widgets, which help in faster development of the applications. Knowing what Streamlit supports natively, can help readers visualize their ideas faster. From creating form-based applications to embedding media, Streamlit widgets help create a fully functional data app in no time.

Chapter 4: Styling and Layouts in Streamlit: This chapter covers layout options provided by Streamlit. Although not a full-fledged web application development framework, Streamlit provides great tools to enhance the design of your data application and apply design principles to build an aesthetic data application. We will discuss Streamlit state management, custom components library, and what it means to design an engaging and well-structured application.

Chapter 5: Data Visualization with Streamlit: This chapter focuses on integrating and creating compelling visualizations with Streamlit. Whether it's simple charts or complex real-time data-driven insights, transforming raw data into visually engaging content can be achieved using Streamlit. We will also explore Streamlit chat widgets and third-party visualization libraries and compare their functionalities and use cases.

Chapter 6: Streamlit and Machine Learning: This chapter dives into integrating machine learning models with Streamlit, showcasing how to build and deploy interactive apps powered by machine learning. Readers will learn how to load pre-trained models, visualize predictions, and make the models accessible to a wider audience.

Chapter 7: Advanced Streamlit Concepts: In this chapter, we explore more advanced features, including caching, secret management, app testing, and optimizing performance. We will also discuss the recommended design patterns and pitfalls, ensuring that the data application.

Chapter 8: Deployment of Streamlit Apps: In this chapter, we will explore the steps required to deploy Streamlit applications using various options like Streamlit Sharing or cloud providers. We will deploy our Product Recommender application using Streamlit Sharing and Google Cloud Platform and compare various considerations to note when deciding on the deployment platform.

Chapter 9-11: Hands-On Projects: Easy-Advanced: In these chapters, we dive into real-world projects from easy to advanced. These projects can help you streamline your learning and explore tooling options available with Streamlit.

Chapter 12: Build and Enhance Your Portfolio: This chapters gives a walkthrough on Streamlit’s active and vibrant global community boards and how to leverage them to enhance your portfolio. From contributing to open-source projects to marketing your apps, you will learn practical tips for getting your work noticed in the broader data and tech community.

Chapter 13: Enhancing Streamlit Development with AI Tools: The final chapter introduces how AI tools can complement Streamlit development. Readers will gain insights into how AI can speed up coding, assist in generating ideas, and even optimize the development process. We will also touch upon the ethical considerations when using AI tools in your projects.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the *Coloured Images* of the book:

<https://rebrand.ly/cfyi5zj>

The code bundle for the book is also hosted on GitHub at

<https://github.com/bpbpublications/Streamlit-Essentials>.

In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

1. Introduction to Streamlit.....	1
Introduction.....	1
Structure.....	1
Objectives.....	2
Challenges of data presentation.....	2
Need for Streamlit	3
Conclusion.....	5
Exercises.....	5
<i>Answers</i>	6
2. Getting Started with Streamlit.....	7
Introduction.....	7
Structure.....	7
Objectives.....	7
Setting up Streamlit.....	8
<i>Comparing installation options</i>	8
<i>Running Streamlit locally using virtual environment</i>	8
<i>Running Streamlit using Docker</i>	8
<i>Running Streamlit using Streamlit community cloud</i>	9
Streamlit core concepts	11
<i>Streamlit app model</i>	12
<i>Exploring the Streamlit hello app</i>	12
Hands-on: Product recommender app.....	16
<i>Project structure</i>	16
<i>Running your first Streamlit app</i>	18
Conclusion.....	20
Exercises.....	20
<i>Answers</i>	22
3. Exploring Streamlit Widgets	23
Introduction.....	23
Structure.....	23
Objectives.....	24

Widgets overview	24
Streamlit core widgets	24
<i>Display widgets</i>	24
<i>Text input widgets</i>	26
<i>Numerical input widgets</i>	27
<i>Date and time widgets</i>	27
<i>Other input widgets</i>	30
<i>Single choice widgets</i>	32
<i>Multiple choice widgets</i>	35
<i>Button widgets</i>	36
<i>Data widgets</i>	37
Streamlit magic	42
Hands-on: Adding widgets to enhance the app	43
<i>Choosing the right widgets</i>	43
Conclusion	47
Exercises	47
<i>Answers</i>	48
4. Styling and Layouts in Streamlit	49
Introduction	49
Structure	49
Objectives	50
Designing engaging and well-structured application	50
User interface design with Streamlit	50
<i>Streamlit layouts and containers</i>	51
<i>Streamlit utility widgets</i>	54
<i>Streamlit control flow widgets</i>	55
<i>Streamlit theming</i>	58
<i>Rendering HTML and CSS</i>	60
State management	62
Streamlit custom components	67
Hands-on enhancing product recommender application	68
Conclusion	74
Exercises	75
<i>Answers</i>	76

5. Data Visualization with Streamlit	77
Introduction.....	77
Structure.....	77
Objectives.....	78
Importance of data visualization in data analysis.....	78
Streamlit visualization options.....	78
<i>Streamlit core chart elements</i>	78
<i>Summarizing the chart types and comparing use cases</i>	83
<i>Streamlit supported charting libraries</i>	84
Streamlit display and aesthetics widgets.....	91
<i>Display progress and status widgets</i>	92
<i>Display message widgets</i>	94
<i>Media widgets</i>	95
<i>Metric widget</i>	96
Streamlit chat widgets.....	97
Hands-on: Integrating visual elements in Product Recommender.....	100
Conclusion.....	114
Exercises.....	115
<i>Answers</i>	116
6. Streamlit and Machine Learning	117
Introduction.....	117
Structure.....	117
Objectives.....	118
Introduction to machine learning in Streamlit.....	118
Setting up your machine learning environment.....	119
<i>Essential Python libraries</i>	119
<i>Preparing your machine learning models</i>	120
<i>Starting with a simple machine learning model</i>	120
<i>Integrating machine learning trained model with Streamlit</i>	121
Dynamically adjusting model hyperparameters.....	122
<i>Understanding model hyperparameters</i>	122
<i>Implementing interactive controls for hyperparameters</i>	122
<i>Tips to improve the performance</i>	124
Hands-on: Integrating ML in Product Recommender.....	125
Conclusion.....	138

Exercises.....	138
<i>Answers</i>	139
7. Advanced Streamlit Concepts.....	141
Introduction.....	141
Structure.....	141
Objectives.....	142
Streamlit advanced topics	142
Callbacks.....	142
Secret management	145
<i>Best practices for Secret Management using Streamlit</i>	147
Working with timezones	147
<i>Best practices when using different timezones in application</i>	147
Building mutating charts with Streamlit.....	148
<i>Best practices when using mutating charts</i>	149
Connecting to databases with Streamlit.....	149
<i>Best practices for database connections</i>	151
App testing with Streamlit.....	151
<i>Best practices for app testing with Streamlit</i>	153
Design patterns pitfalls to avoid for Streamlit	153
<i>Strategies for defining custom classes</i>	153
Conclusion.....	154
Exercises.....	154
<i>Answers</i>	156
8. Deployment of Streamlit Apps.....	157
Introduction.....	157
Structure.....	157
Objectives.....	158
Hosting considerations for Streamlit applications	158
<i>Hosting Streamlit applications</i>	160
Hands-on hosting application with Streamlit Share	160
<i>Debugging tips</i>	163
Hands-on hosting application using GCP cloud run.....	163
<i>Google Cloud Platform</i>	163
<i>Google Cloud Run</i>	163

Deploying the application.....	164
Setting up continuous integration and deployment for the data application.....	176
Getting the metrics and alert information from Cloudrun.....	180
Other Streamlit deployment options	181
Conclusion.....	181
Exercises.....	181
Answers	183
9. Hands-On Projects: Easy	185
Introduction.....	185
Structure.....	185
Objectives.....	185
Streamlit projects abstract: Easy	186
Project 1: Weather dashboard	186
Implementation guide.....	186
Project 2: Basic stock price viewer	186
Implementation guide.....	187
Project 3: To-do list	187
Implementation guide.....	187
Project 4: Image gallery	187
Implementation guide.....	187
Project 5: Movie recommendation system	188
Implementation guide.....	188
Project 6: BMI calculator.....	188
Implementation guide.....	188
Project 7: Simple expense tracker	189
Implementation guide.....	189
Project 8: Currency converter.....	189
Implementation guide.....	189
Project 9: Interactive quiz.....	189
Implementation guide.....	190
Project 10: Random quote generator	190
Implementation guide.....	190
Hands-on: Implementing sample project on the Weather dashboard.....	190
Project structure for Weather data app.....	191
Project structure for currency converter data app.....	193

Conclusion.....	194
10. Hands-On Projects: Intermediate	195
Introduction.....	195
Structure.....	195
Objectives.....	195
Streamlit project abstracts: Intermediate.....	196
<i>Project 1: Multi-page data analysis dashboard.....</i>	<i>196</i>
Implementation guide.....	196
<i>Project 2: Job application tracker</i>	<i>196</i>
Implementation guide.....	196
<i>Project 3: Budget planner with visualizations</i>	<i>197</i>
Implementation guide.....	197
<i>Project 4: Personal fitness tracker.....</i>	<i>197</i>
Implementation guide.....	197
<i>Project 5: Multi-page portfolio website</i>	<i>198</i>
Implementation guide.....	198
<i>Project 6: Task management application</i>	<i>198</i>
Implementation guide.....	198
<i>Project 7: Multi-page product catalog</i>	<i>199</i>
Implementation guide.....	199
<i>Project 8: Sports team management application</i>	<i>199</i>
Implementation guide.....	199
<i>Project 9: Multi-page event planner.....</i>	<i>200</i>
Implementation guide.....	200
<i>Project 10: Multi-page recipe organizer</i>	<i>200</i>
Implementation guide.....	200
Hands-on: Implementing sample project	201
Conclusion.....	207
11. Hands-On Projects: Advanced	209
Introduction.....	209
Structure.....	209
Objectives.....	209
Streamlit projects abstract: Advanced	210
<i>Project 1: Real-time data analytics dashboard.....</i>	<i>210</i>

Implementation guide.....	210
Project 2: Secure financial portfolio manager	210
Implementation guide.....	211
Project 3: Machine learning model deployment	211
Implementation guide.....	211
Project 4: Customizable news aggregator.....	212
Implementation guide.....	212
Project 5: Advanced time series forecasting application.....	212
Implementation guide.....	212
Project 6: Interactive Geographic Information System	213
Implementation guide.....	213
Hands-on: Implementing sample project on news aggregator	213
Conclusion.....	223
12. Build and Enhance Your Portfolio.....	225
Introduction.....	225
Structure.....	225
Objectives.....	226
Streamlit journey and next steps.....	226
Importance of community engagement	226
Key resources and how to engage.....	226
Official forums and discussion boards.....	226
Streamlit community forum.....	227
Streamlit Official App Gallery	230
Streamlit Component Gallery	235
Crafting your professional portfolio.....	236
The idea behind a comprehensive portfolio	237
Leveraging tools for effective documentation	237
Marketing your skills through your portfolio.....	238
Conclusion.....	238
13. Enhancing Streamlit Development with AI Tools	239
Introduction.....	239
Structure.....	239
Objectives.....	239
Introduction to AI-assisted development tools.....	240
Shift from manual coding to AI-assisted development.....	240

<i>Advantages and guardrails of using AI-assisted tools</i>	240
Exploring categories of AI tools	241
<i>Code completion tools</i>	241
<i>AI-powered coding assistants</i>	242
<i>Prompt engineering platforms</i>	242
<i>Bug detection and code review tools</i>	242
<i>Developer bots</i>	243
<i>Comparison of tool category and Streamlit usage</i>	243
Hands-on: Using ChatGPT in Streamlit development.....	244
<i>ChatGPT overview</i>	244
<i>ChatGPT data privacy, intellectual property, ethical usage recommendations</i>	244
<i>Data privacy with ChatGPT</i>	245
<i>Intellectual property guardrails with ChatGPT</i>	245
<i>Other important terms and conditions with ChatGPT</i>	245
Conclusion.....	256
Appendix A: Streamlit Cheat Sheet	257
Objectives.....	257
Streamlit commands	257
Appendix B: Additional Resources and References	261
Introduction.....	261
Streamlit projects	261
User Experience design	262
Future of AI and ethical usage.....	262
Appendix C: Docker 101: Beginner’s Guide to Containers	265
Objectives.....	265
Introduction to containers.....	265
<i>Understanding containers with traveling stage show analogy</i>	266
<i>Docker: The stage crew for your traveling show</i>	266
Docker basics.....	267
Run docker image.....	267
<i>Getting started</i>	268
<i>Cleaning up</i>	269
Building custom images	269
Index.....	273-278

CHAPTER 1

Introduction to Streamlit

Introduction

In the modern digital realm, where data has earned its reputation as the new oil, our interaction with this invaluable resource has been rapidly transforming. The journey from traditional spreadsheets to intricate BI tools and now to intuitive data apps showcases the ever-evolving canvas of data interactivity. This is where Streamlit shines brightly. It is an open-source Python library that is reshaping how we build and visualize data applications.

Streamlit empowers data professionals to transform their intricate analyses and models into dynamic web apps by serving as a bridge between data science and app development. Designed with Python enthusiasts in mind, it eliminates the need for exhaustive knowledge of front-end development, making data app creation accessible and straightforward.

Structure

The chapter discusses the following topics:

- Challenges of data presentation
- Need for Streamlit

Objectives

In this chapter, we will understand the role of Streamlit in a data project, highlighting how it empowers data professionals to create impactful, accessible, and interactive data applications that resonate with a broad audience, including those with minimal technical expertise.

Challenges of data presentation

As businesses and organizations become increasingly data-driven, the gap between complex data analysis and actionable insights for decision-makers and stakeholders becomes more evident.

Data scientists often find themselves at a crossroads when it is time to present their findings, discoveries, or models. It is prudent that the stakeholders are kept in the loop at every project stage to avoid any miscommunication and differences at later stages. Given ahead is an overview of the high-level stages of data projects.

Let us walk through each stage of the project to understand the challenges better:

- **Problem definition, data collection, and preparation:** The project begins with understanding the business problem, defining the scope, and setting clear objectives. This is an important stage of the project; defining correct boundaries and the definition of done is of utmost importance for a successful project execution. This also acts as a guardrail in the later stages of a project when new requirements are brought in. Challenges include dealing with large volumes of data, ensuring data quality, and addressing missing or inconsistent data. Challenges at this stage include aligning the project with business goals and ensuring stakeholder agreement on the project's direction. Data collection involves gathering relevant data from various sources and preprocessing it for analysis. It is important to communicate the findings and data quality issues to the stakeholders, too. It might not seem important to convey at the start, but it helps to set the right expectations from the data.
- **Exploratory Data Analysis (EDA):** Here, data scientists explore the data to understand patterns, anomalies, and relationships. Challenges include making sense of complex data patterns and summarizing key findings in a way that is understandable to non-technical stakeholders.
- **Model building and insight generation:** This stage involves selecting, building, and validating machine learning models, as well as model results, to generate actionable insights. Challenges include translating technical findings into business insights and validating these insights with stakeholders.
- **Deployment and maintenance:** The final model is deployed into a production environment. Challenges include integrating the model with existing systems and ensuring its performance over time.

Each of the stages has its varied technical hurdles. However, one common thing across all the steps is the challenge to constantly communicate and keep the stakeholders in the loop. When communicating with stakeholders who might come from different areas of expertise like sales, advertising, and marketing, it is crucial to ensure that the presentation tools are selected carefully to communicate the key points effectively and precisely.

Visualization serves as a powerful tool in bridging the gap between technical and non-technical stakeholders in such projects. It translates complex data sets into intuitive, easy-to-understand graphical formats, enabling stakeholders who may not have the technical expertise to grasp intricate patterns and insights at a glance. This visual approach not only facilitates clearer communication but also aids in collaborative decision-making. By presenting data in a visually engaging and accessible manner, stakeholders can more effectively comprehend the implications of the data, leading to informed discussions and more efficient decision-making processes. Visualization is a universal language, democratizing access to insights and fostering a shared understanding among diverse teams. Also, it is important to remember that the end goal is always to assist in providing insights to the business.

There are many dashboarding options available in the market today. Most tools cater to specific use cases; some are good at churning out heavy-weight BI insights or building a fancy dashboard, but the fanciest tool might not be the answer here. A simple tool is required that can build faster without over-engineering.

There are some traditional methods like using Jupyter Notebook for building faster, but these might present challenges such as:

- **The complexity of sharing:** Raw data, detailed algorithms, or even Jupyter notebooks with non-technical stakeholders can lead to confusion. It demands an intricate dance of explanations, which might fail to drive the point home.
- **Static visuals:** While visuals like charts and graphs are a step forward, they are static. They showcase a snapshot in time rather than a dynamic flow of data, which might not encapsulate the entire story or the potential variations.
- **Lack of interactivity:** Presenting a model without interactivity means stakeholders cannot experiment with inputs to understand potential outcomes. It becomes a one-way street of information flow, lacking the essential feedback loop.

Need for Streamlit

Turning a data analysis or a machine learning model into an interactive web application required a blend of front-end development skills and a stack of web frameworks. This often meant that data scientists needed to collaborate with web developers or invest time learning these frameworks, which is time-consuming and sometimes cumbersome.

Streamlit fills this void by offering a platform where the usual Python code can magically turn into an interactive web application. Think about when you wanted to share your

analysis or a prototype model with a non-technical stakeholder. Instead of navigating through Jupyter Notebooks or sharing static visuals, Streamlit provides a medium to make these insights dynamic, interactive, and web-ready.

The current industry is full of tools for data visualization and app creation. From Dash by Plotly to Shiny for R, the options are varied. However, Streamlit has carved its niche, especially when fast prototyping is the goal, for the following reasons:

- **Ease of use:** Unlike some platforms that require a deep dive into the nuances of web frameworks, Streamlit offers a simpler Python-centric approach. If you are comfortable with Python, you are already halfway there.

Streamlit's promise is to deliver a functional web application with minimal Python code. Gone are the days when JavaScript, HTML, or CSS were barriers.

- **Interactivity at its core:** While tools like Dash offer interactivity, Streamlit's widget ecosystem is designed to be integrated seamlessly, making the building process intuitive and less cumbersome.
- **Widgets** like sliders, buttons, and text inputs are not just add-ons; they are core to Streamlit. They transform static data insights into dynamic narratives.
- **Speed of delivery:** Rapid prototyping is one of Streamlit's forte. The immediate feedback and live-coding feature ensure a quick transition from code to a visual prototype, a feature not all tools can boast of.

The essence of *what you see is what you get* is embodied in Streamlit's live-coding capability, making prototyping swift and iterative.

When comparing Streamlit with other dashboarding tools in the market, an interesting key feature is the ability to save visualizations as code, which brings several significant advantages over traditional dashboarding tools. When you implement your visualizations as code, there are a few advantages that technical engineers might appreciate:

- **Version control and change management:** By saving visualizations as code, Streamlit allows integration with version control systems like Git. This means any changes made to the dashboard are trackable and reversible, similar to how software development teams manage code changes. This capability is a game-changer for data teams, as it ensures that updates or modifications to the dashboard are documented, can be reviewed, and rolled back if necessary.
- **One point of change:** Streamlit's code-centric approach allows changes to a visualization or dashboard that do not require navigating through complex GUI settings. Instead, updates are made directly in the code. This ensures that there is a single point of change, which is particularly beneficial in maintaining consistency and reducing errors or conflicts that might arise from multiple adjustment points.
- **Collaborative development:** Streamlit's compatibility with standard development tools and workflows makes it an ideal platform for collaborative projects. Multiple team members can work on different aspects of the dashboard simultaneously,